Integrating System Health Management into the Early Design of Aerospace Systems Using Functional Fault Analysis

Tolga Kurtoglu, Stephen B. Johnson, Eric Barszcz, Jeremy R. Johnson, Peter I. Robinson

***Abstract*—This paper introduces a systematic design methodology, namely the Functional Fault Analysis (FFA), developed with the goal of integrating SHM into early design of aerospace systems. The basis for the FFA methodology is a high-level, functional model of a system that captures the physical architecture, including the physical connectivity of energy, material, and data flows within the system. The model also contains all sensory information, failure modes associated with each component of the system, the propagation of the effects of these failure modes, and the characteristic timing by which fault effects propagate along the modeled physical paths. Using this integrated model, the designers and system analysts can assess the sensor suite's diagnostic functionality and analyze the "race" between the propagation of fault effects and the fault detection isolation and response (FDIR) mechanisms designed to compensate and respond to them. The Ares I Crew Launch Vehicle has been introduced as a case example to illustrate the use of the Functional Fault Analysis (FFA) methodology during system design.**

***Index Terms*— Fault Detection Isolation and Response (FDIR), Functional Design, Systems Health Management, Testability Analysis.**

I. Introduction

SYSTEM health management (SHM) is considered a system engineering discipline that includes. "the processes, techniques, and technologies used to design, analyze, build, verify, and operate a system to prevent faults and to minimize their effects." [1] SHM, when implemented successfully, greatly enhance safety, affordability, and maintainability of complex systems.

Manuscript received April 25, 2008.

Tolga Kurtoglu is with the Mission Critical Technologies/NASA Ames Research Center, Moffett Field, CA 94035 USA (phone: 650-604-1738; e-mail: tolga.kurtoglu@ nasa.gov).

Stephen B. Johnson is with NASA Marshall Space Flight Center, Huntsville, AL 35812 USA. (phone: 719-487-9833, e-mail: stephen.b.Johnson@nasa.gov).

Eric Barszcs is with the NASA Ames Research Center, Moffett Field, CA 94035 USA (phone: 650-604-1866; e-mail: eric..barszcz@ nasa.gov).

Jeremy R. Johnson is with the USRA-RIACS/NASA Ames Research Center, Moffett Field, CA 94035 USA (phone: 650-604-0727; e-mail: Jeremy.r.johnson@ nasa.gov).

Peter I. Robinson is with the NASA Ames Research Center, Moffett Field, CA 94035 USA (phone: 650-604-3513; e-mail: peter..i.robinson@ nasa.gov).

An SHM system consists of instrumentation components (sensors, wires, data recorders, etc.), a Fault Detection, Isolation and Response (FDIR) module, diagnostic and prognostic software, as well as processes and procedures responsible for information gathering about a systems' health and corresponding decision-making [2-4]. SHM systems are typically used to accomplish two main goals [5]:

(1) To accurately assess the system health and to pinpoint problems and anomalies. In order to achieve this goal, a SHM system constantly monitors the functional health of a system, and detects, identifies, and isolates faults, and responds to potential problems by enabling system reconfiguration or restoration.

(2) To support the management of the system's off-line maintenance and repair operations. An SHM system achieves this by mapping anomalies to physical components that have failed during operation and by predicting physical components that are likely to fail in the future. These components, referred to as Line Replaceable Units (LRU), are then replaced in order to restore a failed system function.

While it is widely accepted that the accomplishment of these goals are critical for a systems' safety, affordability, and performance, successful implementation of SHM systems remains as a key challenge for various technical reasons.

First, the implementation of SHM systems is fragmented at best. Many aspects of SHM have been implemented in many systems. What is missing is a coherent framework to integrate across instrumentation design, onboard FDIR, ground maintenance, diagnostics, and various fault-related analyses.

Second, SHM functions and designs are traditionally implemented only after system models are built and subsystem designs are fully developed. Such add-on approaches to health management, however, carry a high risk in ensuring system safety and are more expensive and difficult than they need to be. This is largely because the SHM features are band-aids to the already-existing design. With proper design and integration of SHM into the design process, many of these problems and complexities can be prevented. Some single-point failures can be designed out of the system, eliminating some SHM patches altogether, while other SHM features can be designed into the system along with the "nominal design," thus reducing costs by avoiding late design changes and by improving maintenance features required for operations. Including health

management capabilities from the beginning of system design allows the engineers and designers to optimize the application and use of SHM systems, and if integrated early enough, the development of safer, and more reliable system architectures.

This paper introduces a systematic design methodology, namely the Functional Fault Analysis (FFA), developed with the goal of integrating SHM into early design of complex systems. The approach is being applied to Ares I system design and vehicle integration with collaborations across multiple NASA centers and industry leaders.

The basis for the FFA methodology is a high-level, functional model of the system that captures the physical architecture, including the physical connectivity of energy, material, and data flows. The model also contains all sensory information, failure modes associated with each component of the system, the propagation of the effects of these failure modes, and the timing by which fault effects propagate along the modeled physical paths. Once this integrated model is built, the designers and system analysts can assess the capability of the sensor suite of the system to isolate the location of faults, and determine if redundant sensors exist to confirm the existence of a fault. Moreover, these capabilities can be used to assess the sensor suite's diagnostic functionality, and to analyze the "race" between the propagation of fault effects and the FDIR mechanisms designed to compensate and respond to them.

The FFA methodology offers immediate advantages for both the design and the operational phase. During the design phase, FFA provides the ability to:

(1) Assess the effectiveness of the sensor suite to isolate faults to the Line Replaceable Units;

(2) Model the fault effect propagation paths and assess the time latencies along those paths;

(3) Document and analyze the FDIR time response capability in terms of sensor detection capability, sensor confirmation, and the time from fault initiation until detection and confirmation;

(4) Assist design engineers by uncovering problems in design issues across subsystem boundaries.

For the operations phase, the FFA methodology provides:

(5) The diagnostic engine and model that support diagnostic system operations. For Ares I, this includes diagnostic engines used at Kennedy Space Center for launch operations, and several test sites in the vehicle build and integration process. The diagnostic systems isolate the locations of faults, and present to the test operators and maintenance personnel the possible failure modes that could be causing the test anomalies.

In the remainder of this paper, we will use the Ares I system design as a case study to explain the proposed approach and the aforementioned range of capabilities. The organization of the rest of this paper is as follows: Section 2 introduces the case example used throughout the paper – Design of the Ares I Crew Launch Vehicle. Section 3 describes the Functional Fault Analysis methodology and the associated design process. Section 4 presents various analyses that can be performed using the FFA approach. Section 5 summarizes major fault prevention and management tools and technologies employed by the aerospace industry. Section 6 summarizes limitations of the proposed approach and lessons learned. Finally, Section 7 presents concluding remarks and an outlook for future work.

II. CASE EXAMPLE: ARES I CREW LAUNCH VEHICLE

In order to demonstrate the FFA methodology, we'll use the Ares I system shown in Figure 1 as a case example. Ares I is [6]:

"a crew launch vehicle — the rocket that will carry space explorers into orbit. Under the goals of the Vision for Space Exploration, Ares I is a chief component of the cost-effective space transportation infrastructure being developed by NASA's Constellation Program. These transportation systems will safely and reliably carry human explorers back to the moon, and then onward to Mars and other destinations in the solar system.

Ares I is an in-line, two-stage rocket configuration topped by the Orion crew exploration vehicle, a service module and a launch abort system. The launch vehicle's first stage is a single, five-segment reusable solid rocket booster derived from the Space Shuttle Program's reusable solid rocket motor that burns a specially formulated solid propellant. A newly designed forward adapter will mate the vehicle's first stage to the second, and will be equipped with booster separation motors to disconnect the stages during ascent. The second or upper stage is propelled by a J-2X main engine fueled with liquid oxygen and liquid hydrogen.

In addition to its primary mission — carrying crews of four to six astronauts to Earth orbit — the launch vehicle's 25-ton payload capacity might be used for delivering cargo to space, bringing resources and supplies to the International Space Station or dropping payloads off in orbit for retrieval and transport to exploration teams on the moon."

Later in this paper, we'll use models and examples from the Ares I system to illustrate the use of the Functional Fault Analysis (FFA) methodology during system design. These models and examples will be abstracted from the design, not necessarily reflecting the exact current design or performance.

Fig 1. Ares I crew launch vehicle

III. FUNCTIONAL FAULT ANALYSIS (FFA) DESIGN METHODOLOGY

Two overarching goals of the FFA design methodology are to promote life cycle cost savings and system dependability. The most obvious functions the FFA task performs to promote these two major goals are the assessment of the Ares system design to better account for fault behaviors (failure effects), and to automate fault isolation and maintenance procedures for long-term operations.

A less obvious, but equally crucial strategy is to bridge between these two major goals with a dual-use philosophy, such that models, tools, and knowledge used for design analyses would also be re-used during operations. Often what has occurred in prior aerospace systems is that the designers would develop and capture all kinds of knowledge to design the system, and then the operational community would develop models and tools to improve system operability and maintenance. This means that design knowledge has typically been captured twice, once for initial design, and again for operations. The Ares I FFA dual-use philosophy is intended to eliminate the need for operations to recapture the relevant design knowledge, thus providing significant life cycle cost savings.

To achieve these goals, the FFA provides a platform on which information regarding component and sub-system functionality, component, sub-system, and system interactions, component failure modes, and the propagation of failure effects between components can be modeled and integrated into a unified framework. The approach provides a coherent, consistent, and formal schema to capture the relationships between components, their failure modes, and the functionality provided by the components and facilitates the assessment of potential system faults, and their impact on overall SHM performance. The details of the proposed framework are explained in the next section.

A. FFA Framework

Figure 2 represents the relationship of the FFA task to other design tasks.

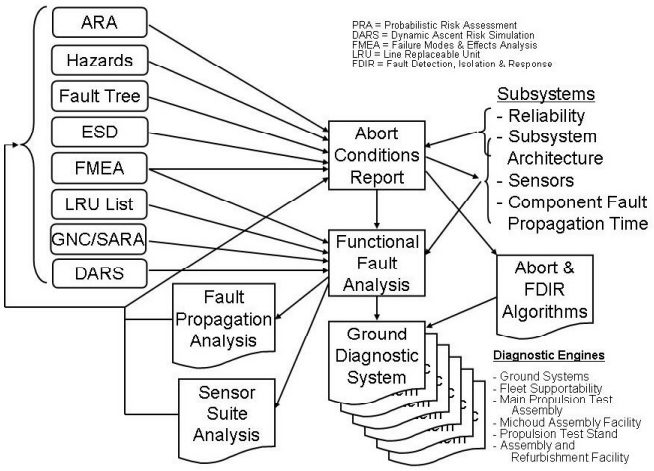

Fig. 2. The FFA framework

Major inputs to the FFA framework include system reference architecture diagrams and technical drawings from sub-system experts, sub-system failure modes and effects analysis (FMEA) and subsequent component reliability data, a list of sensors and proposed sensor locations from systems engineers and integration experts, the line replaceable unit (LRU) list, and component fault propagation timing estimates from sub-system experts, a list of critical conditions that would lead to premature termination of the mission - called the abort conditions, and worst case failure modes for selected abort conditions.

Major outputs from the FFA can be grouped under three themes. Timing analysis outputs (fault propagation analysis in Figure 2) include time-to-effect, time-to-detect, time-to-confirm, time-to-abort-recommendation, and time-to-escape estimates for each failure mode modeled. Testability analysis outputs (sensor suite analysis in Figure 2), on the other hand, encompass results pertaining to sensor suite analysis including first-detection-sensor, confirmation-sensor, list of ambiguity groups and ambiguity statistics, and a list of undetectable failure modes. Diagnostic engine outputs refer to the FFA system model itself as a major output. The FFA task produces a model of the Ares I vehicle that will be exported and reused for operational ground diagnostics.

B. FFA Modeling Environment

This section explains the modeling environment used by the FFA methodology.

The Testability Engineering and Maintenance System (TEAMS) tool suite [7] is selected as the primary platform for modeling. TEAMS includes three tools: TEAMS-Designer, which is the tool to create and analyze the system model, TEAMS-RT, the real-time diagnostic engine that uses a dependency matrix created in TEAMS-Designer to perform real-time fault diagnosis, and TEAMS-Remote Diagnostic Server (RDS) that can “serve” intelligent, optimized diagnostics to thin clients over the Internet or any computer network.

TEAMS is built upon the multi-signal modeling formalism [8], which is a hierarchical modeling methodology where the propagation paths of the effects of a failure are captured using directed graphs. The model is based on structural connectivity or a conceptual block diagram of a physical system connected by signals. Signals describe attributes of system variables to be traced. The so-called “test points” are then added to the model that represent physical locations of sensors and other means used for observing a system. “Tests” are procedures that look at the data from the sensors and make decisions about system attributes associated with those measurements. The test logic can be as simple as Boolean operations on threshold values or may involve complex signal analysis techniques.

This graph topology is then converted into a matrix representation describing the relationship between faults and test points for a given mode of the system. This representation contains the basic information needed to interpret test results

and diagnose failures during on-board monitoring.

The basic advantage of the TEAMS environment is that it enables the integration of certain aspects from traditional functional modeling [9] with the existing modeling and analysis capabilities of the tool suite. First, it provides the modelers a graphical interface that facilitates the construction of a hierarchical model of the structure of a system. Second, the TEAMS "signal" functionality allows the designers to specify energy, material, and data flow connectivity within the system traditionally captured during functional modeling.

This integration allows the experts to define fault propagation effects and fault propagation timing specific to certain physical paths and phenomena. Thus, the TEAMS models are ideally suited for building conceptual level models that can be used to analyze the SHM performance and to define proper FDIR strategies during early concept development. Moreover, the core TEAMS models developed can later be leveraged by reasoning systems to interpret test results in real-time and to assess system health.

C. FFA Design Process

The FFA design process is fundamentally based on the development of subsystem models, which are then integrated into a top-level vehicle representation. This model integrates information regarding the system's functional architecture, system sensors, failure modes of components, and the propagation of fault effects. These are explained next.

1) System Functional Model

The process starts with sub-system reference diagrams, which show the functional connections between components within each of the subsystems modeled. These reference diagrams are translated into sub-system models by replicating the components and their physical connections using the TEAMS representation.

In this representation, which is adopted from traditional functional modeling [9], each component is modeled as a functional unit that operates on certain energy, material, and signal flows. The components perform certain functions on these flows to transform them from an input to a desired output state. For example, a "combustion chamber" component will have input flows for propellant liquids (material), command signals (signal), and output flows for exhaust gases (material), thermal and mechanical energy (energy), and pressure and temperature readings (signal).

Figure 3 shows a snapshot of a conceptual schematic for the Ares engine sub-system, and its functional connectivity model in TEAMS. Once again, nodes in the model represent sub-system components and arcs represent energy, material, and data flow between components. A specific coloring scheme is used to distinguish between different flow types. This "basic subsystem connectivity" constitutes the basis for any further modeling and analysis.

2) Failure Modes of Components

Inside the basic model, failure modes are then constructed

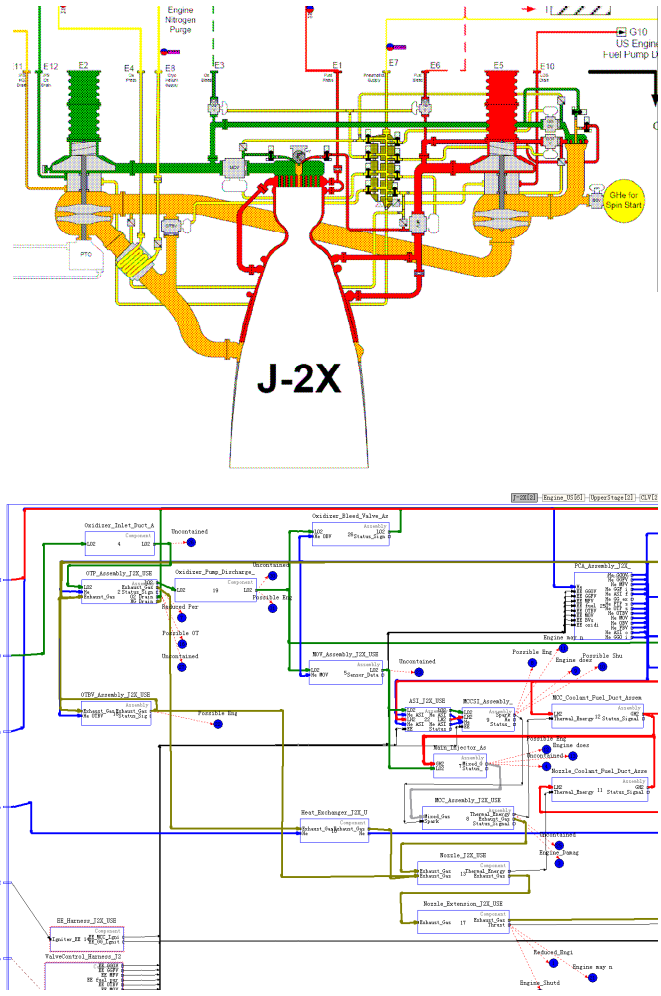

Fig. 3. The engine sub-system schematic (top) provided by the sub-system experts and the corresponding sub-system functional model in TEAMS (bottom)

that correspond to different failure mechanisms for each component. These are taken from the FMEA reports and represented as the lowest level nodes in the FFA model. Figure 4 shows one of the failure modes, FM1 – "external leak", modeled as part of a generic "tank" component.

3) Failure Mode Propagation Paths

After the failure modes of individual components are represented in the FFA model, their potential effects are modeled next. Each failure mode produces a specific effect or set of effects, which propagate along the relevant physical paths (fluid, thermal, electrical, etc.). These effects are modeled using a specific TEAMS feature called "functions," which associates each failure mode effect with every component along the fault propagation path. The path itself is briefly described in the subsystem FMEA, but is formalized by the functional connectivity as represented in the model.

As an example, consider Figure 4 again. For FM1 – "external leak", the failure mode effects propagate along the liquid (material) flow to the next component "line" and along the pressure signal (signal) flow to the component "pressure

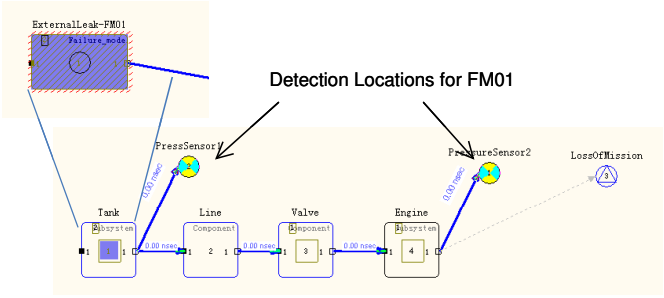

Fig. 4. The failure mode "external leak" for a generic "tank" component in the TEAMS model.

sensor". Specific test point locations, and a potential end effect node (more on these in the next sections) are also shown in the figure. In this way TEAMS can model the LRU, subsystem and system level effects of an FMEA.

4) *Sensors and "Test" Points*

The purpose of a diagnostic system is to isolate the location of a fault and identify the failure mode based on information gathered from sensor readings and tests. Thus the FFA model must represent the location of all sensors in the flight vehicle, and for ground configurations, any other test points from which maintainers might gather information. The sensors are represented as a node just like any other component, except that they are also associated one-for-one with TEAMS "test points." Test points are the mechanism for identifying the location of information-gathering sites, and are also necessary to ensure function propagations, since the TEAMS tool traces from failure modes through components to test points. The two "test points" that can be used to trace the failure mode "external leak" are shown in Figure 4.

5) *Effect Nodes*

Certain downstream failure effects that correlate to nodes of separately-generated fault trees are modeled by creating "effect nodes" in the model. The basic assumption here is that fault effects propagate along relevant physical paths until they reach an effect node, which is an end-result of the fault. Figure 4 illustrates the end effect node "LossOfMission" for the "external leak" failure mode of the tank component.

Using the effect nodes, a system fault tree can automatically be generated and used as a validation tool. That is, the TEAMS-generated fault tree, which hierarchically groups effect nodes, can be compared with the system fault tree developed by hazard analysis groups.

IV. ANALYSES USING THE FFA METHOD

Traditionally, fault analysis is performed only after the system's architecture has been conceived and the components are selected and designed. This is primarily done through Failure Modes Effects and Criticality Analysis (FMECA) [10], which can only be performed after component specifics are available. The major shortcoming of this strategy is that by the time system analysts conduct an FMECA, the system design is mature and many faults are already been designed into the system. Contrary to this traditional approach, FFA provides a way to analyze faults and failures while specific hardware and component selection processes are underway.

In the FFA, this is accomplished through time-to-criticality (the time from initiation of a fault until loss of fault containment—the fault effects cannot be stopped) analysis. The analysis takes the initial system architecture, and accounts for the failure of functions performed by architectural system elements. Since each propagates along paths with certain physical processes, and the connections between architectural components are formally captured, both the failure of components and the propagation of the fault symptoms from the component failures can be analyzed. These determine timing, redundancy, and fault and error containment requirements for the system, and an allocation of health management functions to various system control mechanisms. [1]. Moreover, if the FFA method is implemented early enough during conceptual design, it has the potential of influencing the system architecture design. Thus, decision support through the FFA analysis holds the promise of developing safer and more reliable system architectures.

In the application of the FFA method to the Ares system design, the FFA model is primarily used to perform two analyses: (1) the fault propagation and response timing analysis, and (2) the testability analysis. The former is aimed at assessing time-to-criticality measures in order to define top-level requirements for system "Abort" recommendations, associated algorithms and FDIR strategies. The latter analysis, on the other hand, is used to aid the selection and configuration of sensors. These analyses are explained next.

A. *Fault Propagation and Response Timing Analysis*

The purpose of Fault Propagation Timing and Fault Propagation Response Timing analysis is to assess the "race condition" between the Time to Criticality (TTC) (the time from initiation of a fault until loss of fault containment—the fault effects cannot be stopped), and the Fault Detection, Isolation, and Response (FDIR), which must necessarily be faster than the TTC to mitigate the fault effects. For Ares, one particularly important response is "abort", which for Ares means the transmission of an abort recommendation message to Orion, and in some cases, an automatic Ares response, such as Upper Stage Engine shutdown or closing gaseous hydrogen valves in MPS. This relationship between anomaly detection

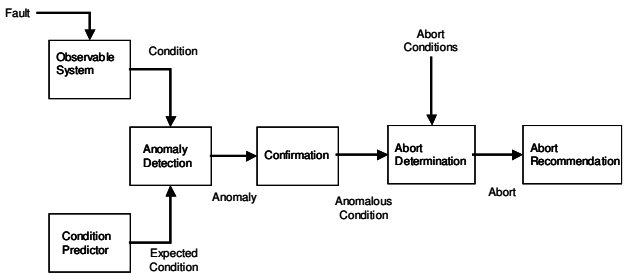

Fig. 8. The relationship between anomaly and abort conditions

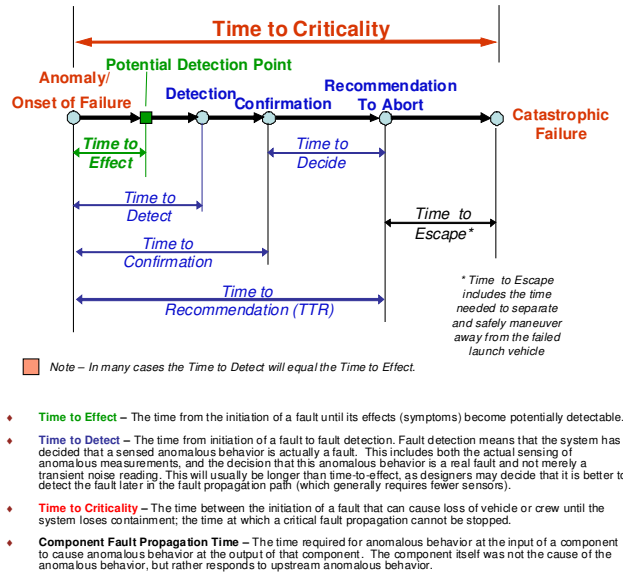

Fig. 6. Timing definitions. These timing values can be assessed using the FFA model

and abort conditions for the Ares system is illustrated in Figure 5.

The FFA model can assess the Time-to-Effect (the time from onset of the failure until its effect(s) are potentially detectable), the Time-to-Detect, the Time-to-Confirmation, the Time-to-Decide, and most parts of the fault effect propagation time (that is, parts of Time-to-Criticality) along the internal vehicle fault propagation paths. The definitions of some of these timing values are given in Figure 6.

The TEAMS-based Ares model is essentially a collection point for timing information gathered by the FFA team from Subsystem Experts and other analysis groups. The TEAMS tool allows for automatic summation and representation of this data once it has been input into the model.

Once a failure mode has been selected for modeling, the fault propagation paths are determined and the physical effect that propagates is modeled with functions. The FFA team extracts the names of all nodes and arcs (connections between nodes) along the fault propagation paths (i.e. those that have the relevant fault effect functions modeled) and inputs those onto a Microsoft Excel spreadsheet. This spreadsheet is given to the Subsystem Experts to fill in the relevant component fault propagation times. Based on the failure mode, the mission phase, and other assumptions relevant to the failure mode (such as size of the leak to be modeled), these times may be based on more or less sophisticated simulations and analyses. Other timing information, including Avionics data processing and transmission times, and Ares vehicle dynamics, are gathered from the Avionics and Guidance and Control Groups, respectively, and input into the model. Once the spreadsheet is filled in, the FFA team inputs the data, and the TEAMS tool can provide a sum of the times along the propagation path.

B. Testability Analysis

The TEAMS tool was built primarily for the purpose of maintenance diagnostics, in which maintenance personnel acquire sensor and other test measurements (often by humans in a troubleshooting role), and based on what they find, to isolate the location and root cause of the problem(s) so identified. During the analysis phase, TEAMS provides various capabilities to address the model's abilities in terms of fault isolation, sensor placement, and fault propagation path analysis.

The simplest analysis is the so-called "Design for Testability (DFT) Analysis", which allows both forward and backward chaining. In forward chaining, the modeler selects a failure mode, and the model automatically propagates the effects of that fault, as defined by the functions that model these effects, to propagate downstream. If the system has been properly designed for testability, one or more test points at sensor locations will detect the fault effect. In the FFA Ares model, each sensor has an associated test point, and it is at the test point that TEAMS allows the modeler to define a set of measurements (tests / sensors) which are compared with the functions activated by the failure mode. In the Ares model, all failure mode effects that lead to abort conditions should be detected by at least two sensors. TEAMS provides detection statistics that sum up the ability of the current sensor suite to detect all failure modes, and a list of all failure modes that are currently not detectable. TEAMS can also determine when there are "redundant tests" (that is, redundant sensors). While TEAMS would typically be used to determine which sensors can be removed with this capability, for Ares Abort Analyses, this is a desired characteristic for the Ares design, to confirm a single sensor reading with a second one.

TEAMS also provides the capability for backward chaining. In backward chaining, the modeler selects a test point. TEAMS determines which functions are monitored at that test point, and then works upstream along the fault propagation path (searches for functions activated at the test point), to determine all possible failure modes that could have activated the sensor (all failure modes with the relevant function that are connected upstream in the network of arcs and nodes). Examples of forward and backward chaining are illustrated in Figure 7.

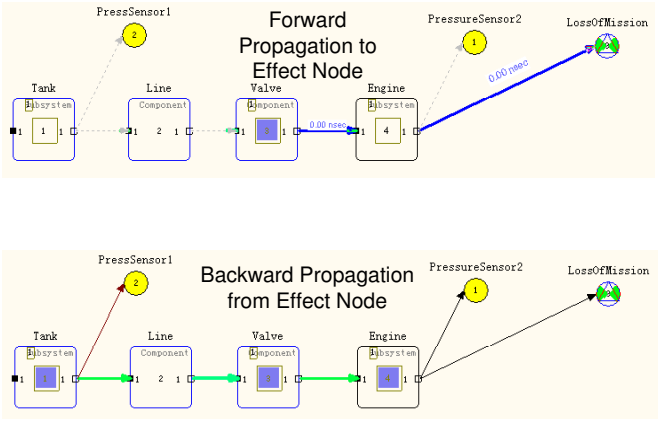

Fig. 7. Forward propagation to an effect node, and backward propagation from an effect node

TEAMS then provides statistics generated from these forward and backward searches. The "ambiguity group analysis" compares the connectivity of all functions to both failure modes and test points. Each ambiguity group is a list of components for which there are insufficient sensors to determine in which of the components in the ambiguity group the underlying failure mode resides in. They also arise if a variety of failure modes give rise to the same fault effect signature. The output of the ambiguity group analysis is the list of all groups of components where the location of a fault cannot be precisely located, and statistics (part of the Testability Figures of Merit (TFOM)) that summarize the total set of ambiguity groups at the selected level.

For the Ares Abort analyses, it is desired that the FFA model be used to perform these statistical assessments for all failure modes identified with each Abort Condition, and for all Abort Conditions. To get accurate statistics, it is necessary to model all relevant failure modes that relate to the abort condition, and also to determine the appropriate set of components and subsystems to include in the analysis.

There are several mechanisms to do this using TEAMS. One mechanism is to select all components and test points related to the particular Abort Condition, and then run the analysis only on those components and test points. This can be done by assigning a "unique technology label" to each module (node) relevant to an Abort Condition, and then running the analysis against the entire set of modules with that technology label. The "technology label" is simply a way of assigning a unique identifier to a group of related model modules. This has the effect of eliminating other modules from the analysis. In practice, this means that all failure modes that lead to the Abort Condition must be analyzed, their propagation paths defined by associated functions, and then all modules that are associated with these functions assigned a unique technology label associated with the abort condition.

One of the major challenges for Ares (and other systems) is whether faults can be isolated to the LRU level, so as to allow for replacement of the unit. If a set of sensors (test points) report anomalous readings, the question is whether these measurements can unambiguously isolate the location of the fault, and secondarily, the specific failure mode that produced the measured effects. TEAMS handles different isolation levels by allowing the modeler to assign a hierarchy label to every node in the model. It provides a pre-defined set, including the "LRU" level and the "failure mode" level, which is by definition the lowest level of the model. When analyses are run, the modeler selects the hierarchy level at which to perform the analysis.

V. RELATED WORK

Fault analysis and management has been a central theme in the design of complex systems. Failure Modes and Effects Analysis (FMEA) [10] has been the most widely used technique for conducting fault and impact analysis. FMEA systematically examines individual system components to assess risk and reliability. It is a bottom-up approach that starts at the component level and follows an inductive logic to determine the consequences of critical component failures. Accordingly, failure mechanisms in which each component can potentially fail are identified and evaluated separately to determine what effect they have at the system level. Complimentary to FMEA, fault tree analysis (FTA) [11] is performed using a top-down approach. FTA starts with identification of a high-level failure event, such as "loss of a vehicle" or "loss of crew". A deductive logic is then followed to drive contributing events that could lead to the occurrence of immediate higher-level events. At the end, the analysis presents the chain of events combined with logical gates in a tree structure. Using this approach, possible event paths from failure root causes to top-level consequences can be captured. Finally, Probabilistic Risk Assessment (PRA) [12,13] is a method that combines a number of fault/event modeling techniques such as master logic diagrams (MLD), event sequence diagrams (ESD) and fault trees (FT) and integrates them into a probabilistic framework to prioritize risk drivers during design. A number of PRA-based methods are developed at NASA as well as in industry including the QRAS [14] and SAPHIRE [15].

Each of these methods has characteristic strengths and weaknesses. As discussed earlier, FMEA is a bottom-up method, which is very accurate at the level of the components for which it generates the failure modes. However, its "effects" analysis becomes progressively weaker as the fault effect propagates further away from the component and into subsystem and system-level effects. FTA is the opposite due to its event-driven nature. It is an outstanding high-level system representation of the major causes of system failure, but becomes progressively weaker as it progresses "down" the fault tree from the systems to the subsystems and below. It lacks failure mode details typically provided by FMEAs. In both FMEA and FTA, the analyst hypothesizes the failure effects (FMEA) and the failure causes (FTA) based on an informal understanding of the system. Often this is accomplished by consultation with the designers and by review of design drawings.

The FFA approach presented here can be thought of as a "middle-out" approach which provides a hierarchical model that can be used to assess failures and their effects using a spectrum of representation levels from failure modes to end effects. Moreover, it formalizes the information traditionally captured separately by FMEA and FTA. Instead of starting from failure modes and hypothesizing the effects, or starting from the failure end-effects to hypothesize about causes, the FFA formally models the architecture as it is created by the designers, the failure modes generated by the FMEA analysts, and the end-effects generated by the FTA analysts. Analysis using the FFA is an experimental method, because the analyst can initiate a failure mode to see how the fault propagates through the architecture (forward chaining), or he can activate an effect node or sensor to determine the failure modes that

could have activated that effect node or sensor. The TEAMS tool can formally produce both the FMEA reports and the Fault Tree. These two products can be compared to the existing FMEAs and Fault Trees to assist with verification of both, along with the design. The strength of the FFA is therefore the analysis of the relationship of failure modes to failure effects through a formally modeled architecture.

Another approach that models fault effects and their propagation is timed failure propagation graphs (TFPG) developed at Vanderbilt University [16]. TFPG models are directed graphs where the nodes represent failure modes, discrepancies, and monitors, and the arcs represent causality. Moreover, likelihood and temporality of causation can be attributed to the arcs of the graph. The TFPG model represents temporal propagation of fault effects and is used for failure diagnosis to describe observed discrepancies in a system.

The basic modeling scheme of the FFA method is function-based. Originally, functional modeling is used as a form-independent method for representing electro-mechanical systems [9,17]. A typical functional model consists of the energy, material and signal flows into and out of a system and the functional descriptions that are performed on these flows. The FFA model deviate from these traditional functional models by replacing textual functional descriptions (such as "transmit EE") with abstract component concepts representing associated functionality (such as "wire"). In that regard, the FFA model resembles a similar graph based representation, called a Configuration Flow Graph (CFG), that is developed by Kurtoglu et al. [18]. A CFG captures conceptual components in a system, their connectivity, and energy, material, and signal flows between them. This representation enables designers to think through the system layout by following the input and output flows through the system components, and model failure modes and their propagation effects and timing associated with system components. Examples of research combining functional modeling and failure analysis for spacecraft design include the function-failure design method, or FFDM [19,20]. This method uses a functional model for a system in combination with historic failure information to map the functionality of a system to potential failure modes. This method has also been proposed to guide the design of ISHM systems, and used in effect to integrate ISHM system functionality design decisions into the design lifecycle [21]. In addition to these tools, Kurtoglu and Tumer developed the Functional Failure Identification and Propagation (FFIP) framework [22] that combines hierarchical system models with behavioral simulation and qualitative reasoning. The method is used to estimate potential functional failures and their propagation paths under critical event scenarios. This method is later extended to include quantifiable measures that define risks based on the role of functionality in accomplishing design goals. Using the extended method, impact analysis results can be related to decision making in order to guide system level design decisions based on functional failures [23].

The FFA methodology presented here extends prior work in this area by providing a systems modeling and integration framework that combines information from functional modeling, FMEA, FTA, and failure modes of sub-system components into a single framework for functional failure and subsequent analyses. One of the main novelties of the FFA methodology is the inclusion of temporal information into the fault assessment and impact analysis domain. In this regard, the FFA modeling approach provides a capability that represents timing information as it relates to functional failures and their propagation.

VI. LIMITATIONS AND LESSONS

Like any other new process, the FFA task has not been problem-free in its implementation. However, by far the most significant issues have been related to organization and process, as opposed to technical limitations.

Initially, the FFA was conceived as a true functional fault analysis, as implied by the name. By this, we mean an analysis done very early in the design based on system functions, as opposed to system components. The intent was to do an architectural assessment of the system to ensure that the architecture was properly structured to respond to faults. This meant an initial set of time-to-criticality estimates across the system, compared with the corresponding FDIR and SHM functions. There was a major difficulty in accomplishing a true function-based analysis: the need to depend on the designers for information about their designs. Early in the development phase of a project, the design primarily exists at a subsystem level, and the system design is essentially an informal integration of these detailed subsystem designs. Thus the most accurate information about the system came from the subsystem designers and their designs. For the most part, subsystem designers do not think about their subsystems in terms of functions, but rather in terms of the actual components they are creating and analyzing. Talking about functions was not effective because subsystem designers do not think this way. Thus the FFA task quickly moved to a generic component basis.

A true function-based analysis is most useful prior to the preliminary design—that is, before specific components are defined. Once a real design exists, the functional basis is not useful, certainly from the perspective of subsystem designers. The implication is that a function-based analysis is useful primarily during early feasibility studies and architectural assessments prior to the existence of a system's preliminary design. However, at this time, one cannot rely on subsystem designers for information, because the detailed information at this level does not exist. The implication is that we must do this analysis based on pre-existing information about typical functions and physical processes associated with those functions. Otherwise, the reliance on designers will push the analysis later in the design process when component designs exist. At this point, there is little benefit in performing a truly function-based fault analysis. In essence, the FFA task that

began in the early preliminary design phase was already too late for a true system-level architectural assessment of fault behavior and its associated architecture.

While the initial FFA work for Ares I was too late for system SHM architecture development, it has proven valuable in the preliminary design process to perform instrumentation and abort analyses, and to work out inconsistencies between subsystems. The very fact of early formal modeling has uncovered a variety of interface and documentation problems that have been identified and fixed much earlier than they would otherwise have been. It has provided sufficient value to make the process and its products worthwhile.

VII. CONCLUDING REMARKS

In this paper, we introduced a new functional fault analysis (FFA) methodology that can be used for integrating SHM into early design of complex systems. The basis for the FFA methodology is a model of the system that captures the physical architecture of the system and the physical connectivity of energy, material, and data flows between system components. Moreover, the method incorporates all sensory information, failure modes associated with each component of the system, the propagation of the effects of these failure modes, and the timing by which fault effects propagate along the modeled physical paths.

The integrated model can be used by system designers and analysts to assess the effectiveness of the sensor suite, to isolate faults, to analyze the time for the effect of component faults to propagate along physical propagation paths, and to determine the time response capability of the fault detection isolation and response (FDIR) mechanisms of the system.

Future work is continuing the FFA modeling and analysis capabilities for Ares I, both for continued abort and FDIR analyses, and for the development of a diagnostic engine to support ground-based diagnostic system operations. Accordingly, the current model will serve as the "core model" for ground diagnostics. Using a consistent model during the design phase through operations will enable continuous verification and validation of the models by the system experts, facilitate model reuse, and increase the confidence in the automated system.

ACKNOWLEDGMENT

REFERENCES

[1] Johnson, S. "Introduction to system health engineering and management in aerospace." 1st Integrated Systems Health Engineering and Management Forum. Napa, CA. November 2005.

[2] Patterson-Hine, A., Narasimhan, S., Aaseng, G., Biswas, G., Pattipati, K., "A Review of Diagnostic Techniques for ISHM Applications." 1st Integrated Systems Health Engineering and Management Forum. Napa, CA. November 2005.

[3] Aaseng, G., Patterson-Hine, A., Garcia-Galan, C., "A Review of System Health State Determination Methods" 1st Space Exploration Conference, Orlando, FL, 2005.

[4] Ofsthun, S., 2002, "Integrated Vehicle Health Management for Aerospace Platforms" IEEE Instrumentation and Measurement Magazine, September 2002.

[5] Zuniga F.A., Maclise, D.C., Romano, D.J., Jize, N.N., Wysocki, P.F., Lawrence, D.P., 2005, "Integrated Systems Health Management for Exploration Systems", 1'st Space Exploration Conference, Orlando, FL.

[6] NASA Fact Sheet, 2007, "Constellation Program: America's Fleet of Next-Generation Launch Vehicles, The Ares I Crew Launc Vehicle", George C. Marshall space Flight Center, Huntsville, Alabama.

[7] QSI, Testability Engineering and Maintenance System (TEAMS) Tool, www.teamsqsi.com.

[8] Deb, S., Pattipati, K.R., Raghavan, V., Shakeri, M., Shrestha, R. "Multisignal flow graphs: a novel approach for system testability analysis and fault diagnosis", IEEE Aerospace and Electronics Systems Magazine, Vol.10, No. 5, pp. 14 -25, 1995.

[9] Pahl, G. and Beitz, W., Engineering Design: A Systematic Approach, Design Council, London, 1984

[10] Department of Defense, "Procedures for performing failure mode, effects, and criticality analysis." MIL-STD-1629A.

[11] Vesely, W. E., Goldberg, F. F., Roberts, N. H. and Haasi, D. F., The Fault Tree Handbook, US Nuclear Regulatory Commission, NUREG 0492, 1981.

[12] Greenfield, M.A. "NASA's Use of Quantitative Risk Assessment for Safety Upgrades". In IAAA Symposium. 2000. Rio de Janeiro, Brazil.

[13] Stamatelatos, M. and Apostolakis, G. "Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners v1.1". 2002, NASA, Safety and Mission Assurance.

[14] Groen J. F., Smidts C., Mosleh A., Swaminathan S. "QRAS: The Quantitative Risk Assessment System", Proceedings of IEEE Reliability and Maintainability Symposium.

[15] SAPHIRE, Systems Analysis Programs for Hands-on Integrated Reliability Evaluation, http://saphire.inel.gov.

[16] A. Misra, J. Sztipanovits, and J. Carnes, 1994, `Robust diagnostics: Structural redundancy approach', in SPIE's Symposium on In-telligent Systems.

[17] Hirtz, J., Stone, R., McAdams, D., Szykman, S., and Wood, K., "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Research in Engineering Design 13(2): 65-82, 2002.

[18] Kurtoglu, T., Campbell, M., "Automated Synthesis of Electromechanical Configurations from Empirical Analysis of Function to Form Mapping". Research in Engineering Design, In Print.

[19] Tumer, I.Y. and R.B. Stone, "Mapping Function to Failure During High-Risk Component Development". Research in Engineering Design, 2003. 14: p. 25-33.

[20] Stone, R.B., Tumer, I.Y. and VanWie, M. "The function-failure design method." Journal of Mechanical Design, 2005. 127(3): p. 397-407.

[21] Hutcheson, R. and Tumer, I. Y., "Function-based Co-design Paradigm for Robust Health Management." The 5th International Workshop on Structural Health Monitoring. Stanford, CA. September 2005.

[22] Kurtoglu, T. and Tumer, I.Y., 2007, "A graph based fault identification and propagation framework for functional design of complex systems," ASME Journal of Mechanical Design, (In print.).

[23] Kurtoglu, T. and Tumer, I.Y., 2008, "A Risk Informed Decision Making Methodology for Evaluating Failure Impact of Early System Designs," ASME Design Engineering and Technical Conference, Brooklyn, NY, In Print.